



19 BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENTAMT

12 Offenlegungsschrift
10 DE 195 34 783 A 1

51 Int. Cl.⁸:
G 06 F 11/20
G 11 C 29/00

21 Aktenzeichen: 195 34 783.8
22 Anmeldetag: 19. 9. 95
43 Offenlegungstag: 14. 11. 98

DE 195 34 783 A 1

Mit Einverständnis des Anmelders offengelegte Anmeldung gemäß § 31 Abs. 2 Ziffer 1 PatG

71 Anmelder:
Siemens AG, 80333 München, DE

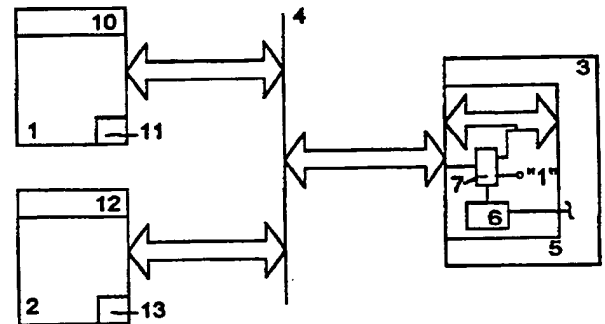
72 Erfinder:
Schwendner, Hans, 83026 Rosenheim, DE

56 Entgegenhaltungen:
DE 31 33 304 A1
US 44 04 647
HÖLSCHER, H. und RADER, J.: Mikrocomputer in
der Sicherheitstechnik, Verlag TÜV-Rheinland, Köln,
1984, S. 7-27 bis 7-30;

Prüfungsantrag gem. § 44 PatG ist gestellt

54 Verfahren zum Selbsttest eines Mikrocontrollers

57 Bei einem Mikrocontroller sind zwei Festwertspeicher (1, 2) mit einander entsprechendem Dateninhalt vorgesehen. Mittels einer Selbsttestroutine wird, beispielsweise durch eine Checksummenüberprüfung, festgestellt, ob die im ersten Festwertspeicher (1) gespeicherten Daten fehlerfrei sind. Im Fehlerfall wird die Adressierungslogik (5) für die Festwertspeicher (1, 2) derart umgeschaltet, daß bei einem Speicherzugriff automatisch auf den zweiten Festwertspeicher (2) zugegriffen wird.



DE 195 34 783 A 1

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen
BUNDESDRUCKEREI 09. 96 602 046/493

5/25

Die Erfindung betrifft ein Verfahren zum Selbsttest eines Mikrocontrollers, der in einem Festwertspeicher gespeicherte Daten enthält. Die Erfindung betrifft außerdem einen Mikrocontroller zur Durchführung des Verfahrens.

Bei Mikrocontrollern ist heutzutage neben dem Mikroprozessorkern meist auch ein Festwertspeicher (Read Only Memory, ROM) auf dem Chip integriert. Der Speicher enthält Daten, die beispielsweise ein Programm darstellen, das durch den Mikrocontroller abgearbeitet wird.

Die vom Festwertspeicher auf dem Chip belegte Fläche ist relativ groß und kann in der Praxis von einem Drittel bis zur Hälfte der Gesamtfläche des gesamten Mikrocontrollers reichen. Die Wahrscheinlichkeit, daß der Festwertspeicher beispielsweise aufgrund von Verunreinigungen während des Herstellungsprozesses oder im Halbleitermaterial fehlerhaft ist, ist folglich relativ hoch. Bereits ein Fehler in einer einzigen, ein Datum speichernden Zelle führt zur Unbrauchbarkeit des gesamten Speichers. Bisher wird der gesamte Mikrocontroller dann als unbrauchbar verworfen oder wird nur eingeschränkt verwendet, indem der Zugriff auf den Festwertspeicher abgeschaltet wird und der Mikrocontroller als speicherlose Produktvariante benutzt wird.

Der Erfindung liegt die Aufgabe zugrunde, ein eingangs genanntes Verfahren und einen geeigneten Mikrocontroller anzugeben, so daß die Verwendbarkeit des Mikrocontrollers erhöht wird.

Erfindungsgemäß wird diese Aufgabe in bezug auf das Verfahren durch die Merkmale des Patentanspruchs 1 gelöst. Ein Mikrocontroller zur Durchführung des Verfahrens ist in Patentanspruch 5 angegeben.

In der Praxis wird die auf dem Mikrocontrollerchip angebotene Speicherkapazität des Festwertspeichers kaum vollständig ausgenutzt. Erfahrungsgemäß wird meist nur die Hälfte oder noch weniger der verfügbaren Speicherkapazität mit Daten belegt. Dies liegt daran, daß die Architektur des Mikrocontrollers und somit auch die verfügbare Speicherkapazität standardgemäß zur Verfügung gestellt wird, während je nach Kundenwunsch bei der Herstellung ein Programm im Festwertspeicher durch einen entsprechend personalisierten Maskenschritt eingebracht wird. Dabei ist der Anwender bestrebt, die verfügbare Speicherkapazität deshalb nur zu dem obengenannten Anteil auszunutzen, um spätere Softwareänderungen und Weiterentwicklungen problemlos mit der gleichen Mikrocontrollerarchitektur realisieren zu können.

Durch die Erfindung wird der nicht genutzte Flächenanteil für den Festwertspeicher zur Bereitstellung von Redundanz verwendet, indem mindestens ein zweiter Festwertspeicher vorgesehen wird, in welchem die Daten des ersten Speichers nochmals entsprechend abgespeichert sind. In praktischen Fällen werden die Daten der beiden Festwertspeicher identisch sein. Gegebenenfalls sind die gespeicherten Daten in den beiden Speichern bis auf solche Unterschiede identisch, die durch die Umsetzung auf den jeweiligen anderen Speicher bedingt sind, z. B. Sprungadressen. Die Feststellung, ob der Dateninhalt des voreingestellten ersten Festwertspeichers fehlerfrei gespeichert ist, läuft zweckmäßigerweise bei der Initialisierung des Mikrocontrollers ab, beispielsweise unmittelbar nach einem "Reset". Während der Abarbeitung des Anwendungsprogramms bleibt aber die durch den mindestens zweiten Festwertspei-

cher vorgesehene zusätzliche Redundanz für den Anwender unbemerkt.

Nachfolgend wird die Erfindung anhand des in der Zeichnung dargestellten Mikrocontrollers näher erläutert. Es sind dabei nur die in Zusammenhang mit der Erfindung wesentlichen Elemente des Mikrocontrollers dargestellt. Der Mikrocontroller enthält neben dem softwaresteuerbaren Mikroprozessorkern weitere Elemente, wie beispielsweise Speicher für Programme und Daten. Hierzu enthält der Mikrocontroller einen ersten als ROM ausgeführten Festwertspeicher 1 sowie einen zweiten entsprechenden Festwertspeicher 2. In ROM 1 und ROM 2 sind Programme und Daten gespeichert, die vom Mikroprozessorteil des Mikrocontrollers während des Normalbetriebs abgearbeitet werden. Die Verarbeitungsvorgänge werden von der zentralen Verarbeitungseinheit (CPU) 3 gesteuert. Die Einheiten 1, 2, 3 tauschen Adreß-, Daten- und Steuersignale über einen internen Bus 4 aus.

Der erste Speicher 1 enthält neben dem Anwendungsprogramm und den Anwendungsdaten auch einen Speicherbereich 10, in welchem die Ablaufsteuerung für das erfindungsgemäße Verfahren abgelegt ist. Durch die CPU 3 ist festgelegt, daß während der Initialisierungsphase des Mikrocontrollers automatisch das im Bereich 10 gespeicherte Programm gestartet wird. Dieses Programm könnte auch in einem sogenannten Boot-ROM-Speicher separat von den gezeigten Speichern 1, 2 abgelegt sein. Ein solches Boot-ROM wird beim Start des Mikrocontrollers automatisch abgearbeitet und enthält entsprechende Initialisierungsroutinen. Der Speicher 1 enthält darüber hinaus einen Speicherabschnitt 11 zur Speicherung mindestens eines Vergleichswerts. Wenn das Überprüfungsprogramm 10 abläuft, wird aus den im Speicher 1 gespeicherten Daten ein aktuell vorliegender Prüfwert berechnet und mit dem im Bereich 11 gespeicherten Referenzwert verglichen. Stimmen beide Werte überein, bedeutet dies, daß die im Speicher 1 gespeicherten Daten korrekt sind. Weichen die Werte voneinander ab, bedeutet dies, daß die im Speicher 1 gespeicherten Daten fehlerhaft sind. Prinzipiell kann für das Überprüfungsprogramm 10 jegliches bekannte Verfahren angewendet werden, mit dem der Dateninhalt des Speichers 1 auf seine korrekte Speicherung hin überprüft wird. Das Programm 10 wird dabei von der CPU 3 abgearbeitet.

Wenn festgestellt wird, daß der Speicher 1 fehlerhaft ist, wird die Adressierungslogik 5 der CPU 3 derart umgeschaltet, daß automatisch aus dem Speicher 2 ausgelesen wird. Zweckmäßigerweise ist auch dort ein dem Überprüfungsprogramm 10 entsprechendes Programm 12 gespeichert sowie mindestens ein entsprechender Vergleichswert 13.

Die Umschaltung zwischen den Speichern 1, 2 wird in der Adressierungslogik 5 dadurch erreicht, daß die auf den Bus 4 ausgegebene Adresse automatisch derart ergänzt wird, daß nicht der Speicher 1, sondern der Speicher 2 adressiert wird. Von den übrigen Teilen der CPU 3 bleibt diese Umschaltung unbemerkt. Beispielsweise kann hierzu das höchstwertige Bit (MSB) anstelle einer "0" (Speicher 1) auf "1" (Speicher 2) umgeschaltet werden. Dies wird dadurch erreicht, daß ein Speichermittel 6 vorgesehen ist, das dann mit einer Kennung entsprechend gesetzt wird, wenn durch das Programm 10 ein Fehler im Speicher 1 festgestellt wird. Das Speichermittel 6, z. B. ein Flip-Flop, steuert einen Multiplexer 7 an, der von der das MSB der von der CPU gelieferten Adresse auf "1" setzt, und somit den Adreßbereich des

Speichers 2 auswählt. Insgesamt bleiben die vorgesehene Redundanz und die gegebenenfalls auszuführende Umschaltung für das Anwendungsprogramm unsichtbar. Das Programm 10 und das Programm 12 werden unmittelbar nach dem Reset während der Initialisierungsphase ausgeführt. Wenn die Programme 10 bzw. 12 nicht vorhanden sind, kann die durch die Speicher 1, 2 zur Verfügung gestellte Speicherkapazität vollständig von einem einzigen Anwendungsprogramm benutzt werden, indem die von der CPU 3 ausgegebenen Adressen den Adreßraum der gesamten Speicherbereiche der Speicher 1 und 2 zusammen umfassen.

Bekanntlich werden integrierte Schaltkreise unmittelbar nach ihrer Herstellung in einem Testautomaten auf Produktionsfehler und Funktionsfähigkeit getestet. Zweckmäßigerweise läßt man auch solche erfindungsgemäßen Mikrocontroller diesen Herstellungstest unbeanstandet passieren, die in nur einem der Festwertspeicher 1, 2 einen Fehler aufweisen. In diesem Fall kann im Speicher 2 das Programm 10 sowie der Vergleichswert 13 entfallen. Sind diese trotzdem vorhanden, können auch nachträgliche Ausfälle durch Alterung erkannt werden. Anstelle der Anordnung des Programms 12 im Speicher 2 kann auch dessen Anordnung, wie bereits angegeben, in einem Boot-ROM vorgesehen sein.

Zweckmäßigerweise bildet das Programm 10 (bzw. 12) eine sogenannte Checksumme über den gesamten Speicherinhalt des Speichers 1 (bzw. 2). Dies bedeutet, daß die bekanntlich wortweise im Speicher organisierten Daten Speicherwort für Speicherwort aufaddiert werden und der Summenwert die sogenannte Checksumme bildet, die dann mit dem gespeicherten Vergleichswert 11 (bzw. 13) verglichen wird. Falls ausreißend, wird eine solche Checksumme nur für einen Teil des Speicherinhalts gebildet oder es werden mehrere Checksummen für mehrere Teile blockweise gebildet. Entspricht die berechnete Checksumme dem gespeicherten Wert 11, bedeutet dies, daß der Speicherinhalt 1 korrekt gespeichert ist. Andernfalls weist dies auf einen Fehler im Speicher 1 hin. Daraufhin wird das Speicherelement 6 gesetzt, so daß das MSB der von der CPU 3 abgegebenen Adressen gesetzt wird und folglich automatisch bei Speicherzugriffen stets auf den Festwertspeicher 2 zugegriffen wird.

Alternativ ist möglich, anstelle der Checksummenbildung für jedes gespeicherte Datenwort außerdem ein Paritätsbit abzuspeichern. Die Überprüfungsroutine 10 berechnet dann für jedes gespeicherte Datenwort das sich ergebende Paritätsbit und vergleicht dieses mit dem abgespeicherten Vergleichsparitätsbit, um festzustellen, ob das Datenwort korrekt vorliegt.

Zur Paritätsbildung werden alle Bits eines Wortes aufsummiert und das sich ergebende LSB gibt das Paritätsbit an. Die Überprüfung mittels Paritätsbits erfordert jedoch im Vergleich zur Checksummenüberprüfung einen höheren Speicheraufwand.

Die Wahrscheinlichkeit, daß im Festwertspeicher 1 mehrere Fehler auftreten, die sich gegenseitig bei der Funktionsüberprüfung durch das Programm 10 kompensieren, ist relativ gering. Um auszuschließen, daß das Programm 10 trotz solcher sich kompensierender Fehler als funktionsrichtig festgestellt wird, ist es zweckmäßig, solche Fehler beim Testen im Testautomaten nach der Herstellung festzustellen und dann den IC zu verwerfen. Andererseits kann auch eine Kombination von Checksummenüberprüfung und Paritätsbitüberprüfung vom Programm 10 ausgeführt werden. Es reicht aus, ein Paritätsbit für eine Anzahl von Datenworten zu berech-

nen und dieses Paritätsbit mit einem im Speicher 1 gespeicherten Vergleichsbit zu vergleichen. Die Wahrscheinlichkeit einer Nichterkennung von sich gegenseitig kompensierenden Fehlern wird dadurch erheblich verringert. Solche eventuell noch auftretenden Fehler können bei dem bereits angegebenen Herstellungstest ausgeschlossen werden oder müssen beim Anwendungstest im System erkannt werden.

Durch das erfindungsgemäße Verfahren wird insbesondere vermieden, daß der Testautomat auf den Chip einwirkt, um die Umschaltung einzuprogrammieren. Die Umschaltung auf den fehlerfreien Speicher wird vielmehr vom Mikrocontroller selbst als notwendig festgestellt und von diesem selbst eingestellt.

Patentansprüche

1. Verfahren zum Selbsttest eines Mikrocontrollers, der mindestens zwei Festwertspeicher (1, 2) enthält, die einander entsprechenden Dateninhalt aufweisen, bei dem durch eine Steuerungseinrichtung (3) des Mikrocontrollers überprüft wird, ob der Dateninhalt in einem der Festwertspeicher (1) fehlerfrei gespeichert ist und bei Vorliegen eines Fehlers Speicherzugriffsmittel (5) derart eingestellt werden, so daß nachfolgend auf den anderen der Festwertspeicher (2) zugegriffen wird.
2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß zur Überprüfung des Dateninhalts des einen der Festwertspeicher (1) mindestens eine Summe aus Teilen des gespeicherten Dateninhalts gebildet wird und diese Summe mit einem Vergleichswert (11) verglichen wird.
3. Verfahren nach Anspruch 2, dadurch gekennzeichnet, daß der eine Festwertspeicher (1) jeweils wortweise adressierbar sind, wobei ein Wort eine feste Anzahl von Bitstellen umfaßt, und daß die Summe über alle Worte gebildet wird und mit dem Vergleichswert (11) verglichen wird.
4. Verfahren nach Anspruch 1 bis 3, dadurch gekennzeichnet, daß für Teilabschnitte des im einen Festwertspeicher (1) gespeicherten Dateninhalts, die mindestens ein Wort enthalten, jeweils eine Paritätsüberprüfung ausgeführt wird, bei der der für jeden Teilabschnitt ermittelte Paritätswert mit einem auf dem Mikrocontroller gespeicherten Vergleichsparitätswert verglichen wird.
5. Mikrocontroller zur Durchführung des Verfahrens nach einem der Ansprüche 1 bis 4, gekennzeichnet durch mindestens einen ersten und einen zweiten einander entsprechende Daten enthaltenden Festwertspeicher (1, 2), eine Steuerungseinrichtung (3), einen Speicher (10), der ein Programm enthält, durch das feststellbar ist, ob der Dateninhalt des ersten Festwertspeichers (1) fehlerfrei ist, und ein Speichermittel (6) zum Speichern einer vom Ablauf des Programms einstellbaren Kennung, in Abhängigkeit derer Adressierungsmittel (5) für Zugriffe auf die Festwertspeicher vom ersten auf den zweiten Festwertspeicher (2) umschaltbar sind.

Hierzu 1 Seite(n) Zeichnungen

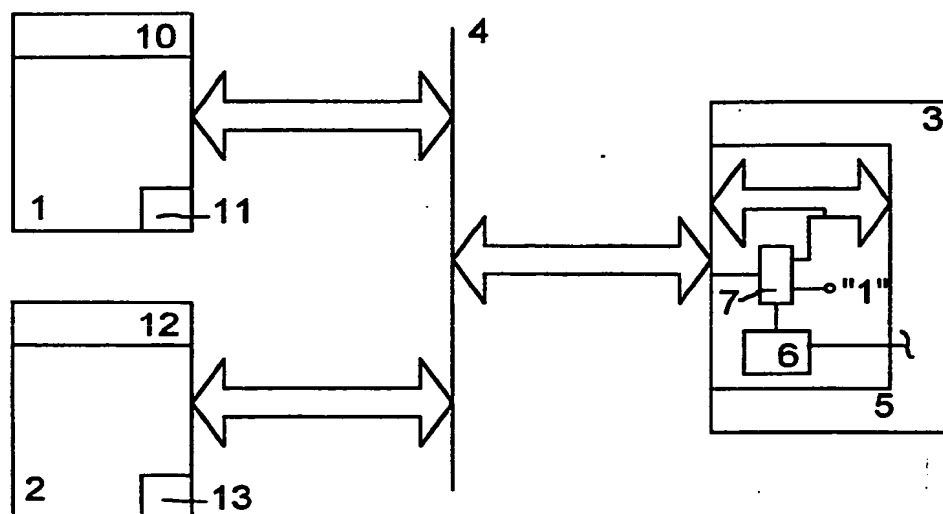


FIG.